

# Examples for Assembler

This section contains examples of using direct Adabas calls in Assembler. The previously defined Adabas files defined are used in each example.

```

*** CONTROL BLOCK
      DS    0F
CB     DS   0CL80      USER CONTROL BLOCK
      DC   CL2' '      RESERVED FOR ADABAS USE
CCODE  DC   CL2' '      COMMAND CODE
CID    DC   CL4' '      COMMAND ID
FNR    DC   H'0'       FILE NUMBER
RC     DC   H'0'       RESPONSE CODE
ISN    DC   F'0'       ISN
ISNLL  DC   F'0'       ISN LOWER LIMIT
ISNQ   DC   F'0'       ISN QUANTITY
FBL    DC   H'100'     FORMAT BUFFER LENGTH
RBL    DC   H'250'     RECORD BUFFER LENGTH
SBL    DC   H'50'      SEARCH BUFFER LENGTH
VBL    DC   H'100'     VALUE BUFFER LENGTH
IBL    DC   H'20'      ISN BUFFER LENGTH
COPT1  DC   CL1' '      COMMAND OPTION 1
COPT2  DC   CL1' '      COMMAND OPTION 2
ADD1   DC   CL8' '      ADDITIONS 1
ADD2   DC   CL4' '      ADDITIONS 2
ADD3   DC   CL8' '      ADDITIONS 3
ADD4   DC   CL8' '      ADDITIONS 4
ADD5   DC   CL8' '      ADDITIONS 5
CTIME  DC   F'0'       COMMAND TIME
UAREA  DC   CL4' '      USER AREA
*
*
*** USER BUFFER AREAS
FB     DC   CL100' '    FORMAT BUFFER
RB     DC   CL250' '    RECORD BUFFER
SB     DC   CL50' '     SEARCH BUFFER
VB     DC   CL100' '    VALUE BUFFER
IB     DC   CL20' '     ISN BUFFER
* * *

```

---

## Example 1

- Find the set of records in file 2 with XB = 99.
- Read each record selected using the GET NEXT option.

### Issue Open Command

```

EXMP1 MVC  CCODE,=C'OP'          OP COMMAND
      MVC  RB(4),=C'ACC.'        ACCESS ONLY REQUESTED
      CALL ADABAS,(CB,FB,RB)     CALL ADABAS
      CLC  RC,=H'0'              CHECK RESPONSE CODE
      BNE  EX1ERR                BRANCH IF NOT 0

```

## Issue Find Command

MVC	CCODE,=C'S1'	FIND COMMAND
MVC	CID,=C'S101'	NONBLANK CID REQUIRED FOR
*		IDENTIFICATION OF THE LIST
MVC	FNR,=H'2'	FILE 2
MVC	ISNLL,=F'0'	ALL QUALIFYING ISNS DESIRED
MVC	IBL,=H'0'	ISN BUFFER NOT REQUIRED
MVI	FB,C'.'	NO READ OF DATA STORAGE
MVC	SB(7),=C'XB,3,U.'	SEARCH CRITERION
MVC	VB(3),=C'099'	SEARCH VALUE
CALL	ADABAS,(CB,FB,RB,SB,VB)	CALL ADABAS
CLC	RC,=H'0'	CHECK RESPONSE CODE
BNE	EX1ERR	BRANCH IF NOT 0
CLC	ISNQ,=F'0'	CHECK NUMBER OF ISNS FOUND
BE	EX1EXIT	BRANCH TO EXIT IF NO ISNS FOUND

## Read Each Qualifying Record

EX1B	MVC	CCODE,=C'L1'	READ COMMAND
	MVC	ISN,=F'0'	BEGIN WITH 1ST ISN IN LIST
	MVI	COPT2,C'N'	GET NEXT OPTION TO BE USED
	MVC	FB(3),=C'RG.'	ALL FIELDS TO BE RETURNED
EX1C	CALL	ADABAS,(CB,FB,RB)	CALL ADABAS
	CLC	RC,=H'0'	CHECK RESPONSE CODE
	BE	EX1D	BRANCH IF RESPONSE CODE 0
	CLC	RC,=H'3'	CHECK IF ALL RECORDS READ
	BE	EX1EXIT	BRANCH IF YES
	B	EX1ERR	BRANCH TO ERROR ROUTINE
EX1D	.	.	PROCESS RECORD . . .
	B	EX1C	BRANCH TO READ NEXT RECORD

## Error Routine

EX1ERR	EQU	*	
*			DISPLAY ERROR MESSAGE
*	.		TERMINATE USER PROGRAM

## Issue Close Command

EX1EXIT	MVC	CCODE,=C'CL'	CLOSE COMMAND
	CALL	ADABAS,(CB)	CALL ADABAS
	CLC	RC,=H'0'	CHECK RESPONSE CODE
	BNE	EX1ERR	BRANCH IF NOT 0

## Example 2

- All records in file 1 are to be read in physical sequential order.
- Each record read is to be updated with the following values:
  - Field AA = ABCDEFGH
  - Field AB = 500
- User is to have exclusive control of file 1.

## Issue Open Command

```

EXMP2 MVC  CCODE,=C'OP'           OPEN COMMAND
        MVC  RB(6),=C'EXU=1.'     EXCLUSIVE CONTROL REQUESTED
        CALL ADABAS,(CB,FB,RB)    CALL ADABAS
        CLC  RC,=H'0'             CHECK RESPONSE CODE
        BE   EX2A                 BRANCH IF RESPONSE CODE 0
        B    EX2ERR              BRANCH IF NOT 0

```

## Issue Read Physical Sequential Command

```

EX2A MVC  CID,=C'L201'           NONBLANK CID REQUIRED
        MVC  FNR,=H'1'           FILE 1 TO BE READ
        MVC  ISN,=F'0'           ALL RECORDS TO BE READ
        MVC  FB(3),=C'GA.'       VALUES FOR FIELDS AA AND AB
*                                           (GROUP GA) TO BE RETURNED
EX2B MVC  CCODE,=C'L2'           READ PHYS. SEQ.
        CALL ADABAS,(CB,FB,RB)  CALL ADABAS
        CLC  RC,=H'0'             CHECK RESPONSE CODE
        BE   EX2C                 BRANCH IF RESPONSE CODE 0
        CLC  RC,=H'3'             CHECK FOR END-OF-FILE
        BE   EX2EXIT             BRANCH TO EXIT IF END-OF-FILE
        B    EX2ERR              BRANCH TO ERROR ROUTINE

```

## Update Record

- The same fields are to be updated as were read.
- The same CID and format buffer can be used for the update command.
- The ISN of the record to be updated is already in the ISN field as a result of the L2 command.

```

EX2C MVC  CCODE,=C'A1'           UPDATE COMMAND
        MVC  RB(8),=C'ABCDEFGH'   VALUE FOR FIELD AA
        MVC  RB+8(2),=PL2'500'   VALUE FOR FIELD AB
        CALL ADABAS,(CB,FB,RB)  CALL ADABAS
        CLC  RC,=H'0'             CHECK RESPONSE CODE
        BE   EX2B                 BRANCH TO READ NEXT RECORD

```

## Error Routine

```

EX2ERR EQU *
*           .                     DISPLAY ERROR MESSAGE
*           .                     TERMINATE USER PROGRAM

```

## Close User Session

```

EX2EXIT MVC  CCODE,=C'CL'       CLOSE COMMAND
        CALL ADABAS,(CB)        CALL ADABAS
        CLC  RC,=H'0'             CHECK RESPONSE CODE
        BNE EX2ERR              BRANCH IF NOT 0
        . . .

```

## Example 3 : User Session with ET Logic

During user session initialization, the user program is to display information indicating the last successfully processed transaction of the previous user session.

For each user transaction, the user program is to

- accept from a terminal 8 characters of input to be used as the key for updating files 1 and 2; and
- issue the Find command for file 1 to determine if a record exists with field AA = input key.

If no record is found, the user program is to issue a message. If a record is found, the user program is to

- delete the record from file 1; and
- add a new record to file 2: Field RA = input key entered.

Other fields are to contain a null value.

If the record cannot be successfully added, the user program is to issue a BT command and display an error message.

If both updates are successful, the user program is to issue an ET command.

## Session Initialization

The section of the program illustrated is only executed during user session initialization:

### Issue Open Command

The OP command is issued with ET data of the previous session being read:

```

EXMP3 EQU  *
      MVC  CCODE,=C'OP'           OPEN COMMAND
      MVI  COPT2,C'E'            ET DATA TO BE READ
      MVC  ADD1,=C'USER0001'     USER IDENTIFICATION
      MVC  ADD3,=C'PASSWORD'     USER PASSWORD
      MVC  RB(8),=C'UPD=1,2.'    FILES 1 AND 2 TO BE UPDATED
      CALL ADABAS,(CB,FB,RB)     CALL ADABAS
      CLC  RC,=H'0'              CHECK RESPONSE CODE
      BE   EX3A                  BRANCH IF RESPONSE CODE 0
      CLC  RC,=H'9'              CHECK FOR RESPONSE CODE 9
      BE   EXMP3                  BRANCH TO REPEAT OPEN
      B    EX3ERR                 BRANCH IF NOT 0 OR 9
EX3A  EQU  *
      CLC  CID,=F'0'             CHECK IF ET DATA FROM
*                                     PREVIOUS SESSION EXISTS
      BE   EX3B                  BRANCH IF NO ET DATA
*                                     . . .

```

### Display ET Data

Display the ET data contained in the record buffer on the terminal screen to inform the user of the last successfully processed transaction of the previous user session:

```

      B    EX3C                  BRANCH TO BEGIN TRANS. PROCESS.
EX3B  EQU  *

```

## No ET Data Received

If no ET data was received, a message is displayed indicating that no transactions were successfully processed during the previous user session.

## Transaction Processing

This section is executed for each user transaction:

```
EX3C EQU *
* . . . ACCEPT INPUT FROM TERMINAL . . .
```

## Issue Find Command

Issue the Find command for file 1 to determine if a record exists with the field AA equal to the input key entered:

```
EX3D EQU *
MVC CCODE,=C'S4' FIND WITH HOLD COMMAND
MVC CID,=C' ' ISN LIST NOT TO BE SAVED
MVC FNR,=H'1' FILE 1
MVC ISNLL,=F'0' ALL QUALIFY. ISNS TO BE RETURNED
MVI FB,C'.' NO READ OF DATA STORAGE
MVC SB(3),=C'AA.' SEARCH CRITERION
MVC VB(8),INPUT SEARCH VALUE
CALL ADABAS,(CB,FB,RB,SB,VB,IB) CALL ADABAS
CLC RC,=H'0' CHECK RESPONSE CODE
BE EX3E BRANCH IF RESPONSE CODE 0
B EX3ERR BRANCH TO ERROR ROUTINE
EX3E EQU *
CLC ISNQ,=F'0' CHECK NUMBER OF RECORDS FOUND
BNE EX3F BRANCH IF RECORD FOUND
```

## Issue Message if No Record is Found

If no record is found, the user program issues a message requesting a correction:

```
B EX3C RETURN TO ACCEPT USER INPUT
```

## Delete Record from File 1

The ISN of the record to be deleted is already in the ISN field and in hold status as a result of the S4 command.

```
EX3F EQU *
MVC CCODE,=C'E4' DELETE COMMAND
CALL ADABAS,(CB) CALL ADABAS
CLC RC,=H'0' CHECK RESPONSE CODE
BE EX3G BRANCH IF RESPONSE CODE 0
CLC RC,=H'9' CHECK IF CURRENT TRANS. HAS BEEN
* BACKED OUT BY ADABAS
BE EX3D IF YES, BRANCH TO REPEAT S4
B EX3ERR BRANCH TO ERROR ROUTINE
```

## Add a New Record to File 2

```

EX3G EQU  *
MVC  CCODE ,=C'N1'          ADD NEW RECORD
MVC  FNR ,=H'2'            FILE 2
MVC  FB(6) ,=C'RA.'        VALUE BEING PROVIDED FOR RA
MVC  RB(8) ,INPUT          VALUE FOR FIELD RA
CALL  ADABAS ,(CB,FB,RB)    CALL ADABAS
CLC  RC ,=H'0'            CHECK RESPONSE CODE
BE   EX3I                 BRANCH IF RESPONSE CODE 0
CLC  RC ,=H'9'            WAS TRANSACTION BACKED OUT?
BE   EX3D                 IF YES, RETURN TO REISSUE TRANS.

```

## Unable to Add a New Record

If the attempt to add a new record is not successful, the transaction is backed out and the user is notified that an error condition exists.

```

MVC  CCODE ,=C'BT'          BACKOUT TRANSACTION
CALL  ADABAS ,(CB)          CALL ADABAS
CLC  RC ,=H'0'            CHECK IF RESPONSE CODE 0
BE   EX3H                 BRANCH IF 0

```

## Backout Not Successful

When the backout is not successful, a message is issued indicating that result.

```

B    EX3ERR                 BRANCH TO ERROR ROUTINE
EX3H EQU  *

```

## Backout Successful

When the backout is successful, a message is issued indicating that after an error was detected, the transaction was backed out.

```

B    EX3ERR                 BRANCH TO ERROR ROUTINE

```

## Updates Successfully Executed : Issue ET Command with ET Data

When the updates have been successfully executed, an ET command with ET data is issued.

```

EX3I EQU  *
MVC  CCODE ,=C'ET'          END OF TRANSACTION COMMAND
MVI  COPT2 ,C'E'            ET DATA TO BE WRITTEN
MVC  RB(8) ,INPUT          ET DATA CONSISTS OF INPUT KEY OF THIS TRANSACTION
CALL  ADABAS ,(CB,FB,RB)    CALL ADABAS
CLC  RC ,=H'0'            CHECK RESPONSE CODE
BE   EX3C                 IF RESPONSE CODE 0, RETURN TO RECEIVE INPUT FOR
*                           THE NEXT TRANSACTION
*
CLC  RC ,=H'9'            CHECK IF CURRENT TRANSACTION HAS BEEN BACKED OUT
*                           BY ADABAS
*
BE   EX3D                 IF CURRENT TRANSACTION HAS BEEN BACKED OUT,
*                           RETURN TO REISSUE TRANSACTION

```

**Error Routine**

```
EX3ERR EQU  *
*          .          NONZERO RESPONSE CODE RECEIVED
*          .          DISPLAY ERROR MESSAGE
*          .          TERMINATE USER PROGRAM
          . . .
INPUT DS CL8          KEY ENTERED FROM TERMINAL
```